Superlative TINs

(with Notes Pages)

Bob Chamberlain

JPL IT Symposium

4 November 2002

TIN is short for "triangulated irregular network", which is a piecewise planar model of a surface. If properly constructed, a TIN can be more than 30 times as efficient as a regular triangulation.

In our project (a ground combat simulation to support U.S. Army training exercises), the TIN is used to represent the Earth's surface and is used primarily to determine whether line of sight is blocked by terrain. High efficiency requires accurate identification of ridgelines with as few triangles as possible.

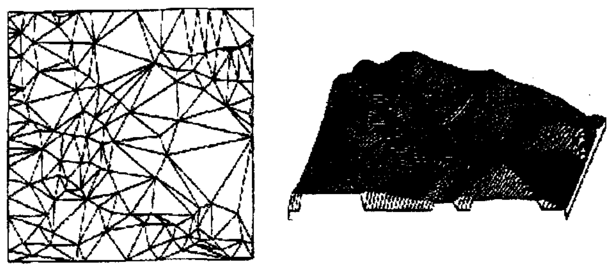
The work currently in progress is the implementation of a TINning process that we hope will produce superlative TINs. This presentation describes that process.

We will also make production runs on a dozen or so playboxes and deliver the software to our customer for further use. Eventually, we will publish a research paper detailing the tradeoffs among the various control parameters and assessing the TINs' actual quality.

California Institute of Technology

What is a TIN?

A Triangulated Irregular Network
is a
piecewise planar model
of a
surface



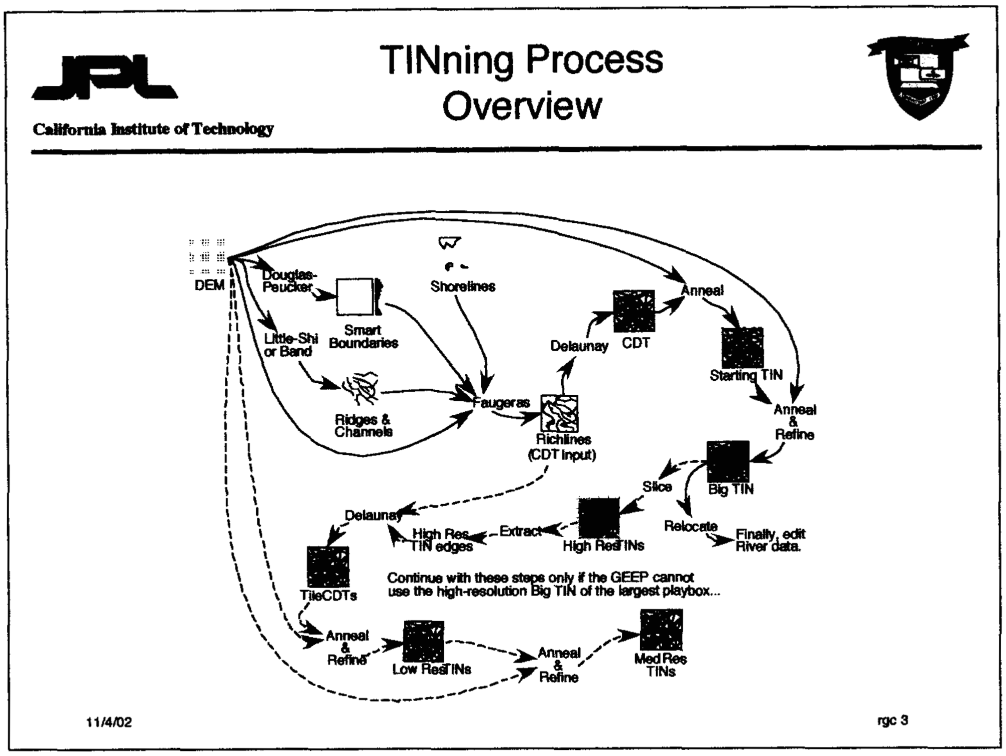
11/4/02 rgc 2

There is not only no known "optimal" technique for fitting a piecewise planar model to a surface, the problem has not even been solved for finding the best segmented line to approximate a general curve on a plane.

There is, however, a generally accepted heuristic for the simpler problem, which is used for "line simplification" on maps. We use it for that purpose and to describe the elevation profile of the edges of our surface. Surface approximation techniques are still an active area of research.

To some extent, what makes one technique better than another depends on the application. In many applications, speed of construction is paramount. In ours, however, the TINs are generated offline, then used over and over to check whether two points are connected by an unbroken line of sight. For this application, the best fit has the fewest edges between an observer and a target, subject to an accuracy requirement. This is almost the same as seeking the smallest number of triangles that provides the required accuracy. There is also a limit on how many triangles can be held in the computer memory.

The fit is also used to compute slopes for mobility determination, but that requirement is much less demanding.



"Ground truth" is a Digital Elevation Model (DEM), an array of elevation values spaced one second (20–30 m) apart, provided by the National Imaging and Mapping Agency and based on data from the Shuttle Radar Topography Mission.

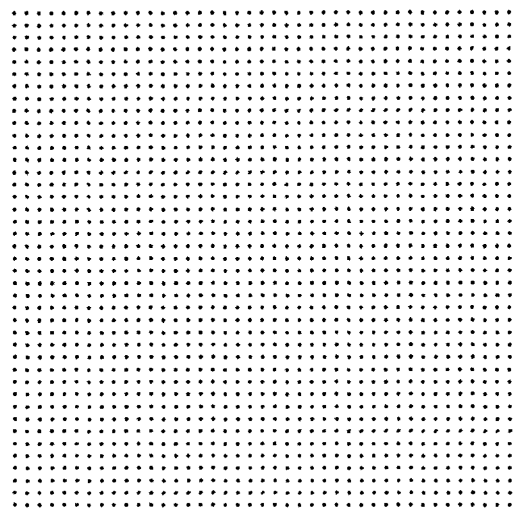
Many applications find it convenient to work directly with the DEM and to assume a regular triangulation that uses all of the elevation posts to interpolate between them. A DEM, however, is a very inefficient way to describe a surface, and many applications, such as military simulations, cannot afford the inefficiency. A TIN may require as little as 3% as much data to achieve the same accuracy. (This number is one of the targets of this study.)

Early TINning efforts were very labor-intensive, using manual identification of structural lines (such as ridges and rivers) and manual insertion of edges to complete the triangulation. Current automated approaches generally start with a few simple triangles and go directly for iterative refinement, splitting the triangles containing the DEM post farthest from the TIN-so-far until an accuracy requirement is met. (In practice, however, a triangle budget is often encountered first.) These automated techniques often produce many "slivery" triangles as an undesirable side effect.

Since accurate determination of line of sight is the most important criterion in our application, the structural lines are especially useful, so we attempt to deduce their locations before constructing the initial TIN. "Slivers" can cause anomalous behavior, so we use a (constrained) Delaunay triangulation as a starting point. The Delaunay triangulation, however, does not use elevation information, so it is thoroughly "annealed". Finally sliveriness and several other quality metrics are considered during refinement in addition to the distance between DEM posts and the TIN surface.

California Institute of Technology

DEM
Digital Elevation Model



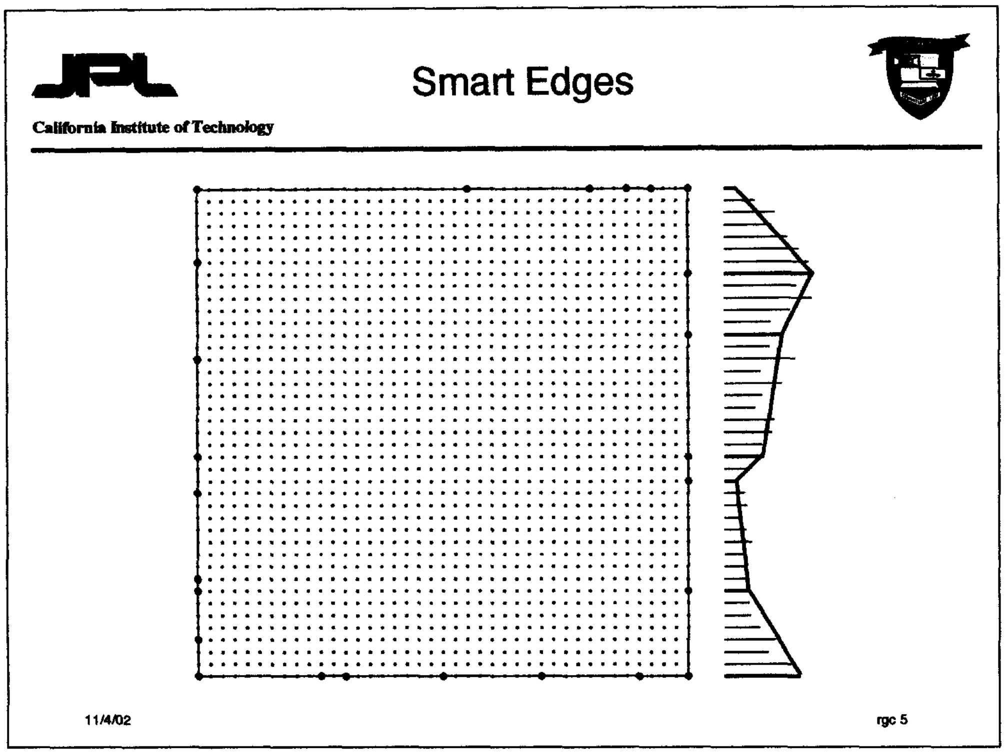
11/4/02 rgc 4

This is the fundamental set of input data. It consists of a regular array of elevation values obtained from NIMA, the National Imaging and Mapping Agency, derived from the SRTM (Shuttle Radar Topography Mission), with post spacing of 1 second of arc (about 30 meters) covering an entire playbox. The array is rectangular in latitude and longitude, but not necessarily square.

This VG shows only 41 x 41 = 1681 posts. Actual post density and extent will be much higher than in these illustrations. In fact, each degree will have 3600 x 3600 = 12,960,000 posts, and our largest playbox has 264 tiles, which implies approximately 3.42 x 10^9 posts.

Because the earth is round and this grid is rectangular, with constant spacing in angular measure (seconds) between points, the spacing in meters in the east-west direction is less for points farther from the equator than for those closer to the equator. Thus, NIMA reduces the spacing between elevation posts at several critical latitudes. DTED1 files (which have 3 second spacing) below 50° have 1200 line segments (actually, 1201 *points*) along the edges of 1° x 1° tiles. From 50° to 70°, they have 600. From 70° to 75°, 400. From 75° to 80°, 300. From 80° to 90°, 200.

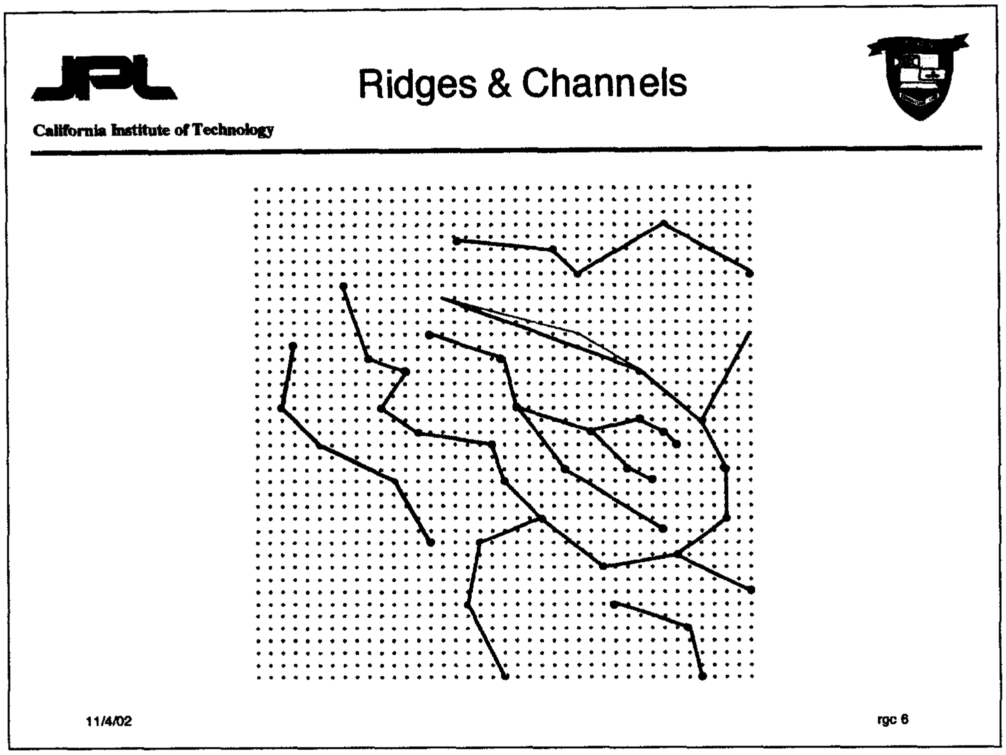
From this data, we extract elevation profiles on the boundaries, terrain breaklines, and — perhaps — shorelines.



The Douglas-Peucker simplification algorithm [Douglas & Peucker 1973] is applied to the DEM data along each edge to select points that represent the profile more efficiently than subsampling the raw DEM data. If necessary, these boundaries will be used to abut mixed-resolution TINs.

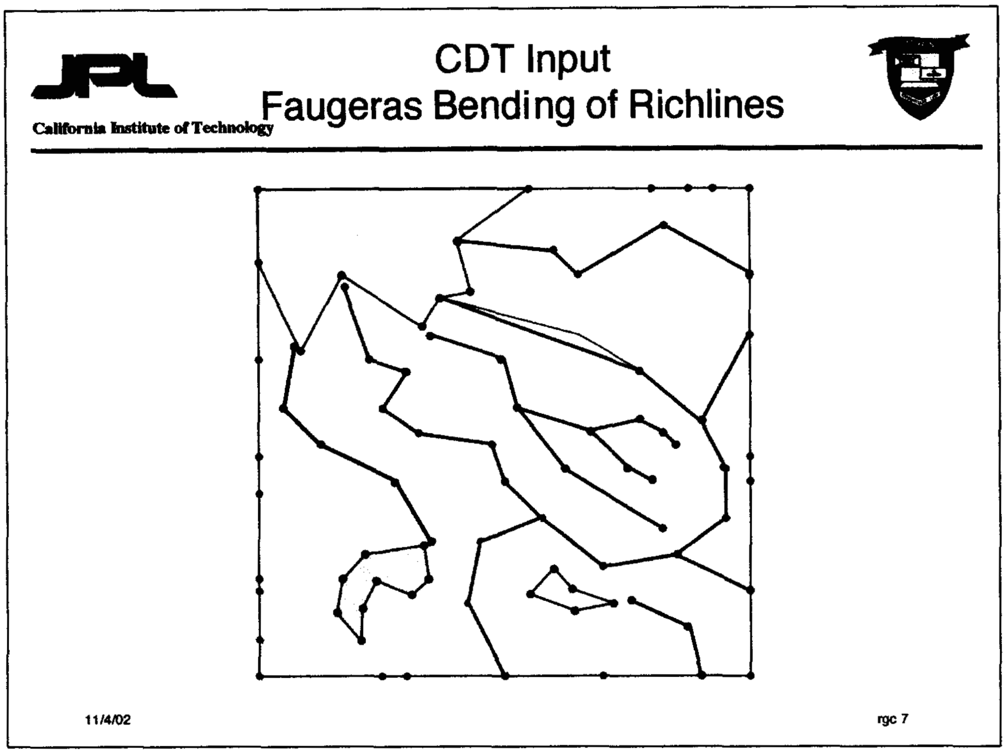
[Whyatt & Wade 1988] give a clear recursive description of how the algorithm works. This algorithm is useful in other steps of the process, as well, but applied in plan view (or three dimensionally), rather than in cross section. Consequently, we wrote a three-dimensional version of the algorithm, with separate tolerance parameters for vertical and horizontal errors.

The TIN refinement algorithm often tries to break an existing TIN edge into two pieces, so we allow for subsequent "unsimplification" of lines during that process.



Ridges and channels are deduced from the DEM. Three algorithms are being considered. The simplest [Peucker & Douglas 1975] uses local tests to identify pits, peaks, passes, ridges, ravines, and breaks by considering the elevation of a DEM post relative to its eight neighbors, but has been observed to produce many extraneous points. Another [Jenson & Domingue 1988] (or [Band 1986]) follows gradients to identify watersheds. The third [Little & Shi 2001] fits a higher-order surface to the DEM, then finds the lines of maximum curvature. We may find it desirable to use divides and channels (from the watershed algorithm) and ridges and cliffs from the high-curvature algorithm.

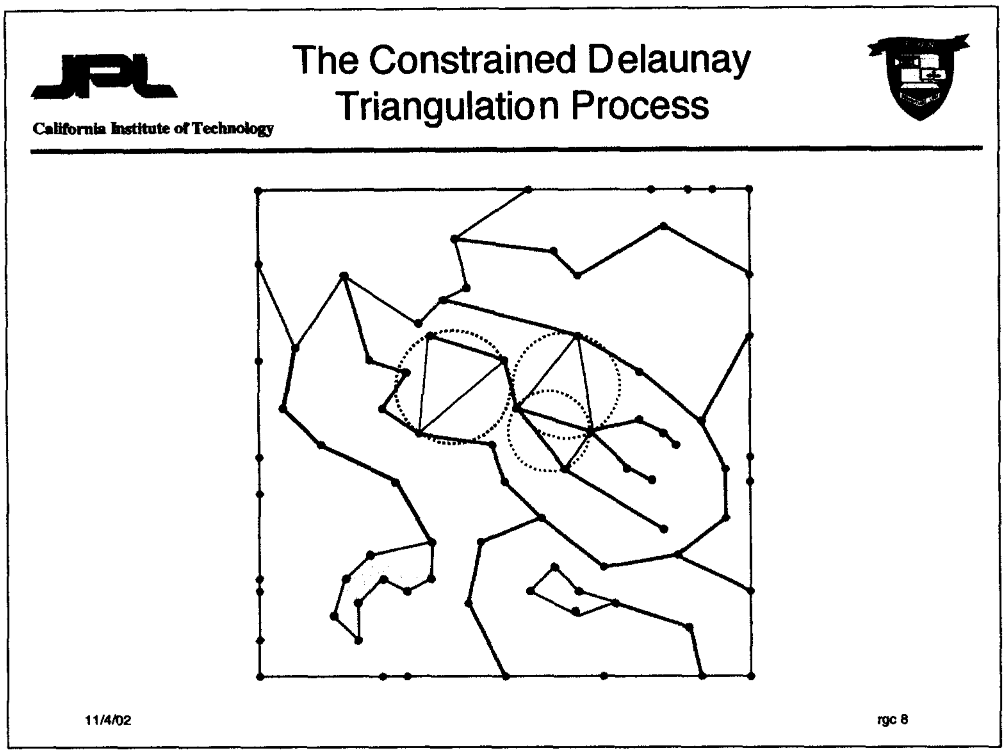
An inevitable part of any of those processes is simplification (by the same Douglas-Peucker algorithm as is used on boundary elevation profiles) of the deduced ridges and channels. This figure suggests the presence of the list of auxiliary points by showing a two-piece, thinner, brown line that might have resulted from simplification.



The remaining input needed to construct the TIN is the shoreline polygons, which represent coastlines and lake boundaries. While the corners of these polygons, if obtained from VMap1 datasets, do not necessarily fall exactly on the DEM posts, we have realized that horizontal resolution is limited by the DEM post spacing in any case and will place them at the nearest DEM locations. This will also avoid the creation of slivers to accommodate constrained locations that are very close together.

To prepare for the next step, the lines that are too long to produce a Delaunay triangulation will be bent (possibly merely segmented) by applying the algorithm given in [Faugeras 1993], using the auxiliary points and Douglas-Peucker simplification. If there are no auxiliary points (as will always be the case for unconstrained edges), the line is split at the DEM post suggested by Faugeras' algorithm.

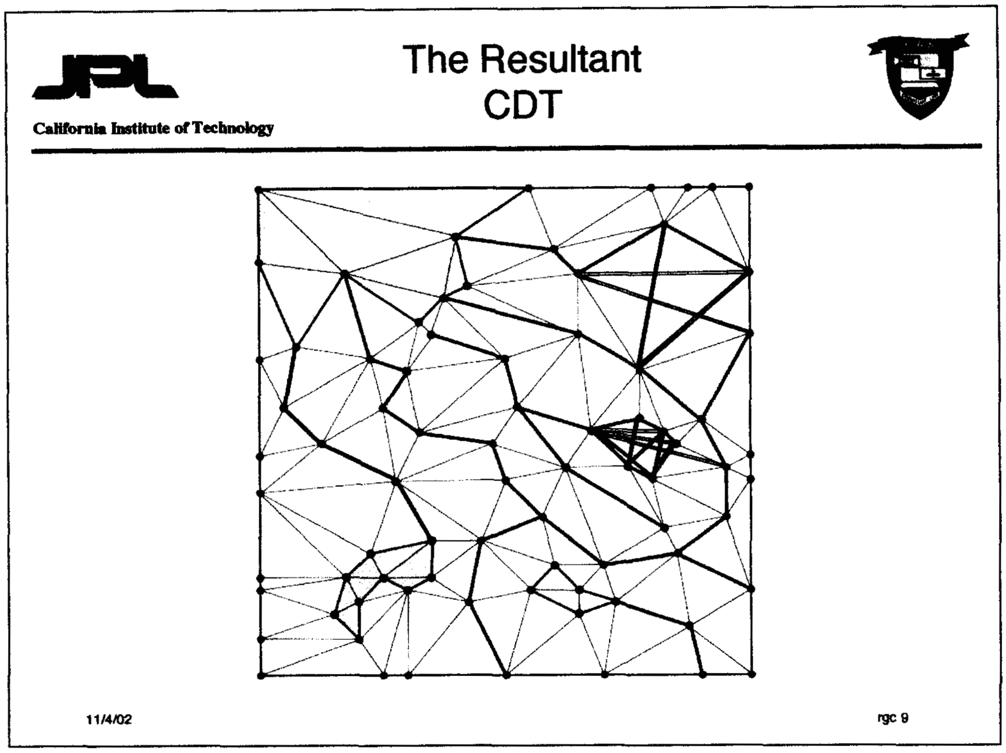
The result is shown in the next VG, which assumes the bend shown here was indeed needed..



The term *richlines* comes from [Douglas 1986] and is so appropriate that I have chosen to use it here.

These richlines are used as constraints in an algorithm to produce a constrained Delaunay triangulation. A Delaunay triangulation has several "nice" properties: Since it is the dual of a Veronoi polygon, its edges connect the constrained posts that are nearest neighbors. Also, it is the triangulation in which the smallest angle is as large as it can possibly be.

Another property of the Delaunay triangulation is used to find the triangles: The circle that circumscribes a Delaunay triangle contains none of the other points in its interior. So a circle through the endpoints of each constrained edge and each other vertex is constructed, then inspected for contents. Although this is inherently an $O(n^3)$ algorithm, it can be made to run in $O(n)$ time (I think) by limiting the search for candidate vertices to those that are nearby and the search for interlopers to the box that bounds the circle.

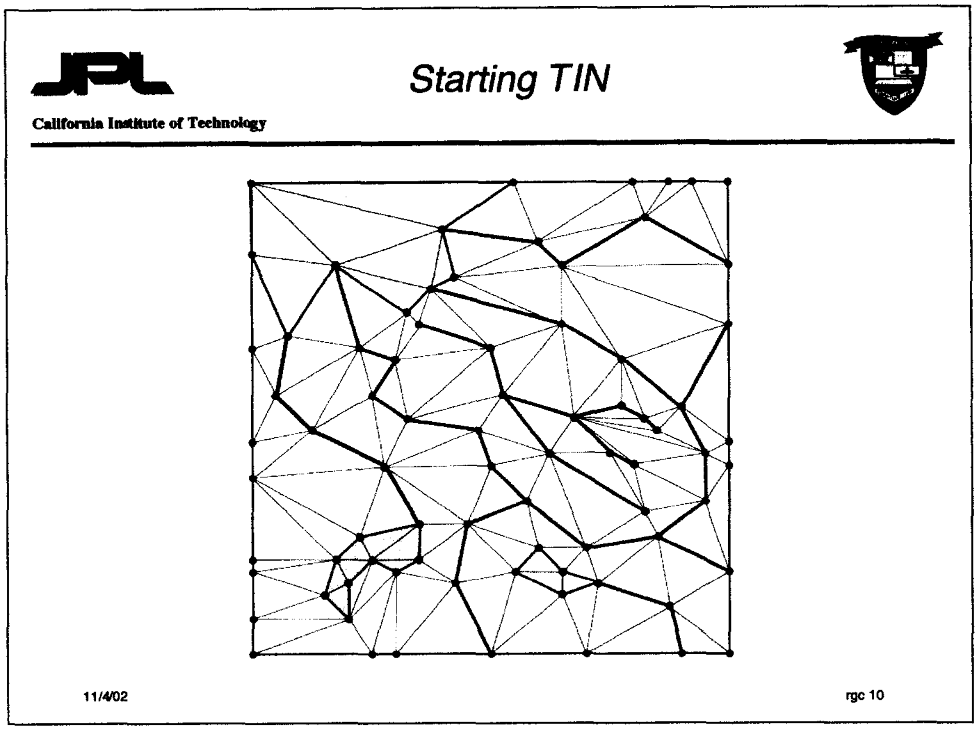


This shows the resultant Constrained Delaunay Triangulation. It does not illustrate the lists of auxiliary points that will still be carried for each of the *unflippable* richlines.

The Delaunay properties do not guarantee the best fit to the terrain, as the elevation data was used only indirectly when finding the unconstrained edges. The doubled lines (in red) illustrate some suspicious Delaunay edges. The final step in producing the Starting TIN is, therefore, determining whether any of the unconstrained edges should be replaced by the other diagonals in their quadrilateral. That process is called *annealing*, and uses an expanded set of quality metrics:

- Maximum difference between the DEM and the TIN
- Sum of squares of differences
- Root mean square difference
- Sliveriness

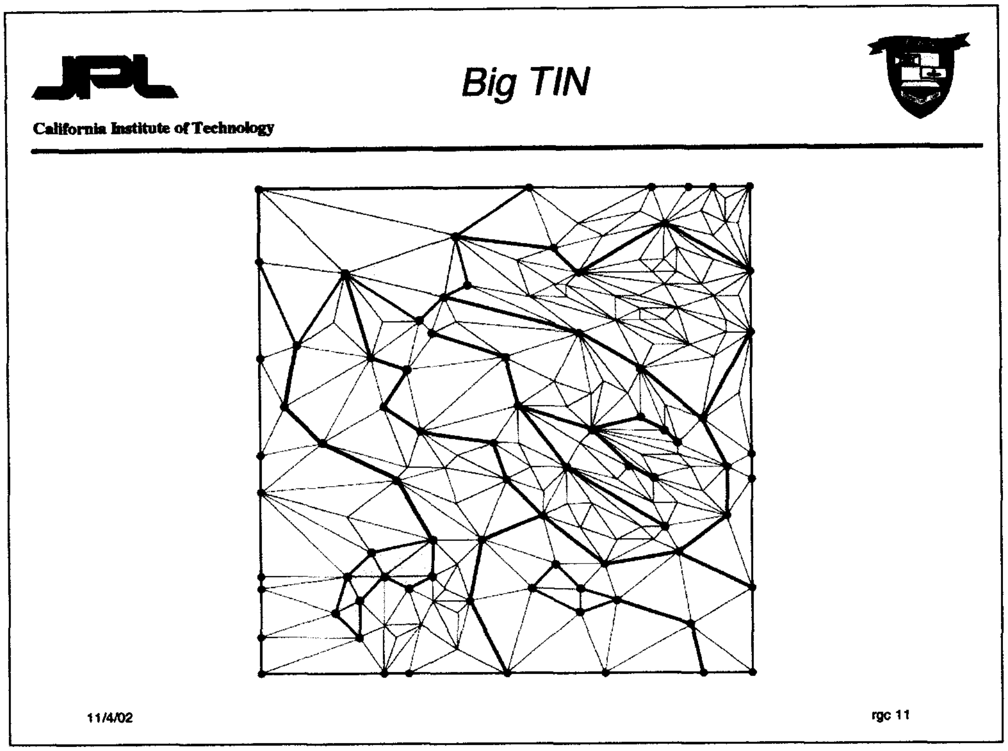
The procedure: References to the triangles will be put into separate lists, one for each metric, sorted according to quality. The lists will be processed round-robin until some "good enough" criterion is met: The worst triangle will be taken off the top of the list and each of its edges will be considered for flipping. When an edge is flipped, the two new triangles are assessed and put in their places in the lists.



The "suspicious edges" in the CDT on the previous page contains unconstrained edges that connect the headwaters of the channels in the basin near the middle of the map. There must be ridges between channels, so I presumed that the annealing process would choose to flip those edges when I constructed the example above. I expect extensive flipping can lead to changes that are as dramatic as shown, where some slivers are much better fits to the terrain than is achieved by the Delaunay triangulation.

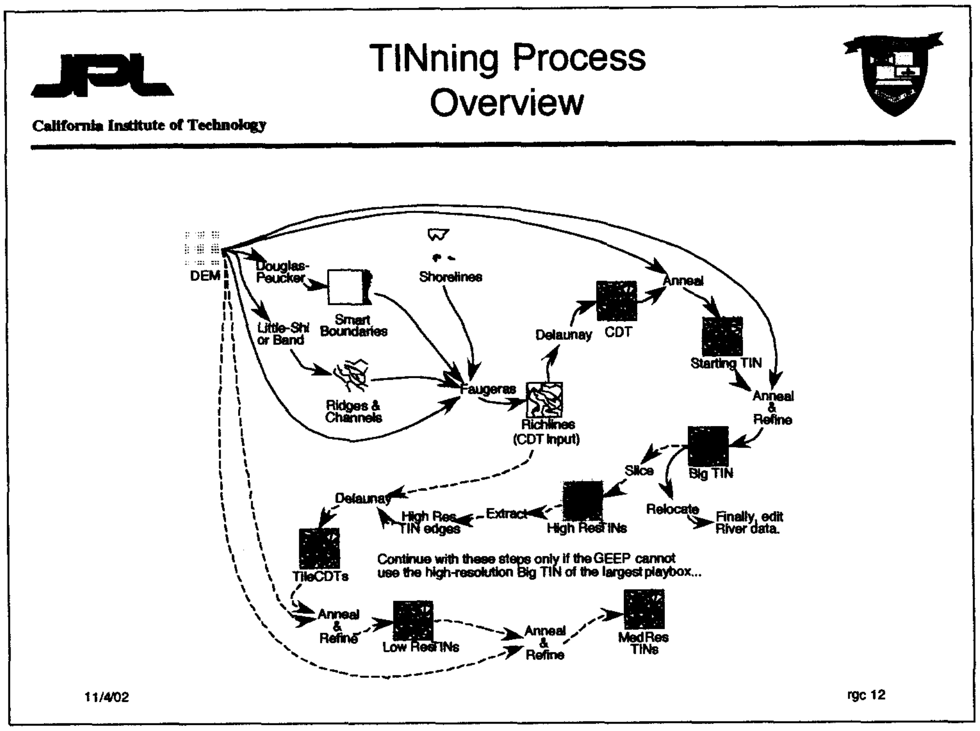
Refinement can now be quite fast, provided the priority queues of triangles can be accessed expeditiously. In the annealing process, the number of triangles is not changed. During refinement, the "worst" triangle at each step and one of its neighbors are split into four triangles.

After all of the quality metrics are satisfied — or, more likely, the triangle budget has been reached — the TIN is subjected to extensive annealing. The result is the high-resolution *Big TIN*, as suggested by the next VG.



If the simulation hardware is not able to deal with a high-resolution TIN of the entire playbox in one gulp, it will have to be sliced into more manageable bites.

Either the *Big TIN* or the tile-by-tile *High-Resolution TIN*s can be used to edit the river network database (which is not needed for TIN construction) so that rivers are located in the deduced channels. One consequence of this editing will be that they will never be modeled as flowing uphill.



If the simulation is not able to handle the amount of data in the high-resolution *Big TIN*, it will be necessary to generate low- and medium-resolution TINs for smaller pieces that tile the playbox. Seamless edges are produced by slicing the *Big TIN*, then using the high-resolution triangles as part of the starting information when reTINning the tiles.

If we find that an entire playbox is impracticably large, we will TIN smaller pieces from scratch, relying on the generation of Smart Boundaries to ensure there are no discontinuities (unexpected cliffs) on the tile boundaries — though changes in slope would be expected.